

1/6

[illegible]

FIG. 1

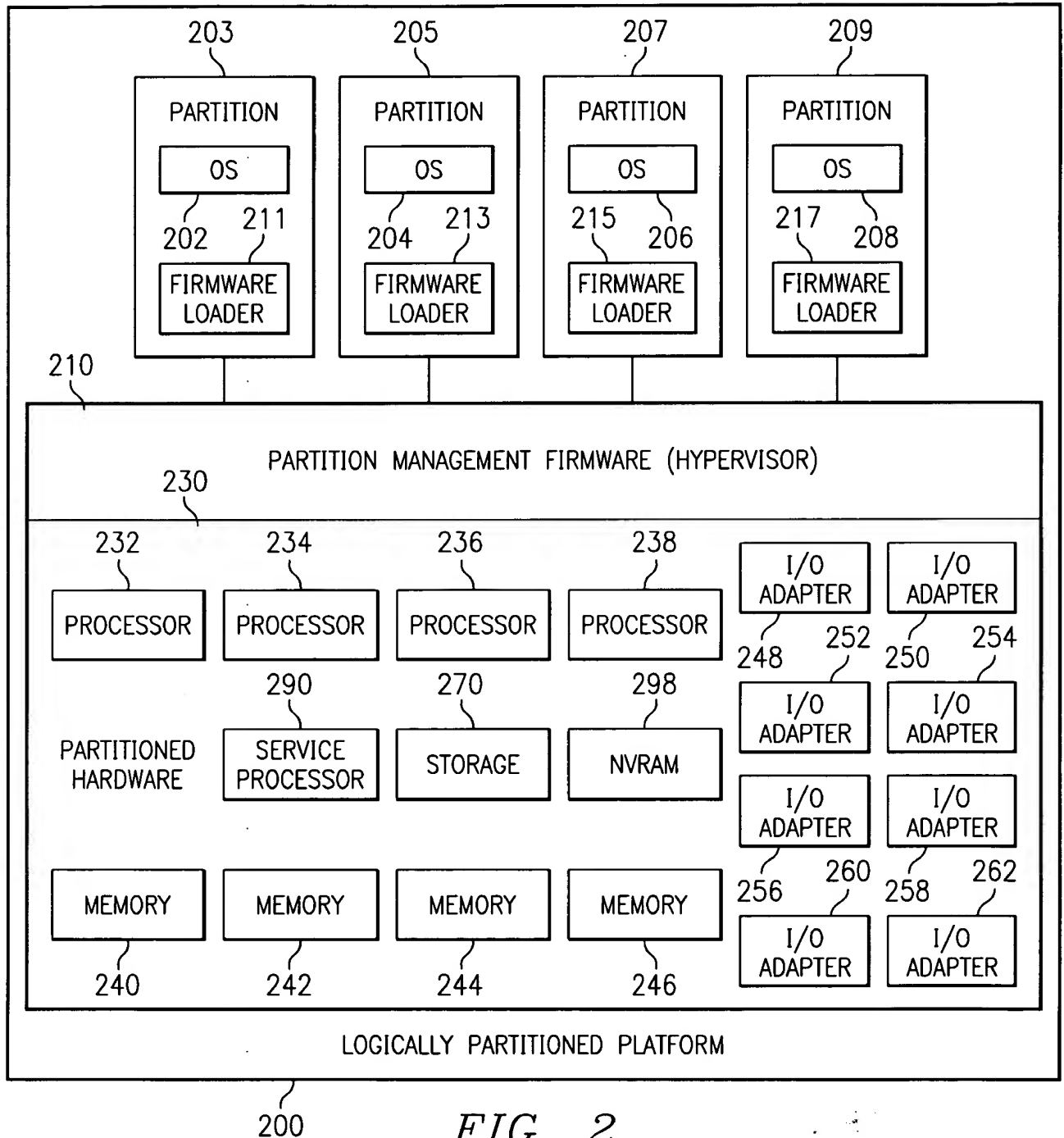


FIG. 2

FIG. 3

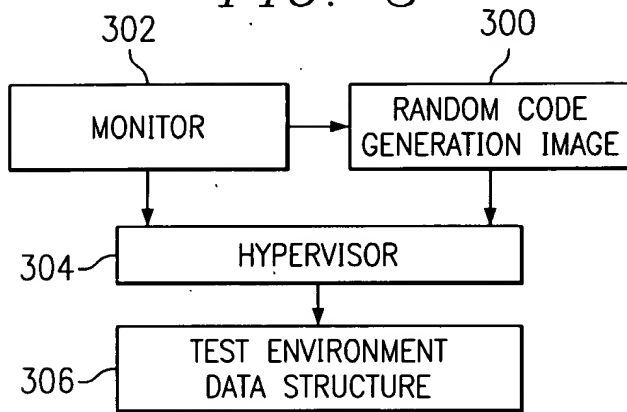


FIG. 4

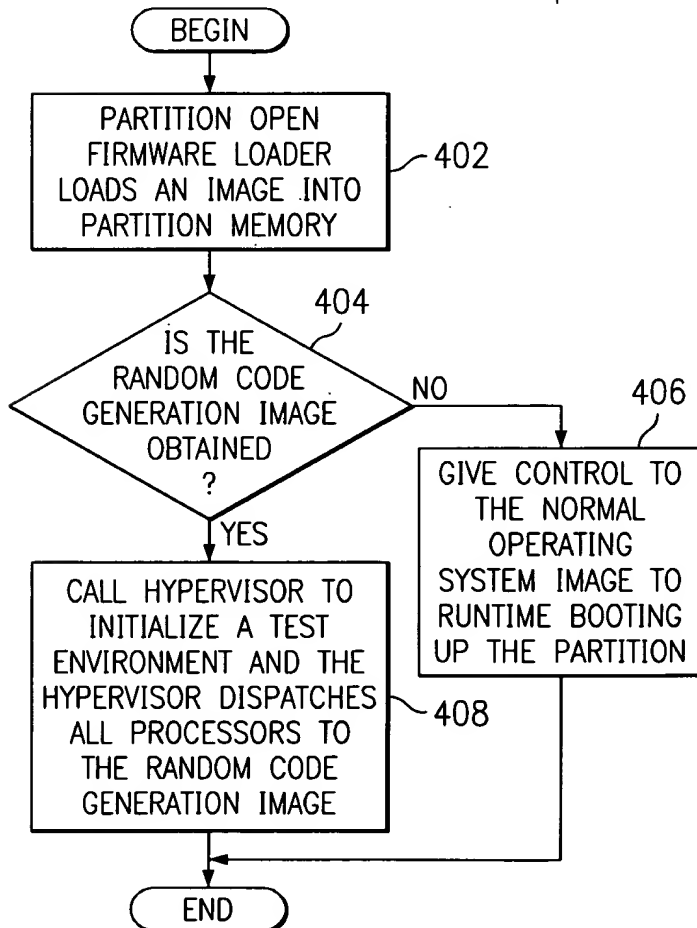


FIG. 5

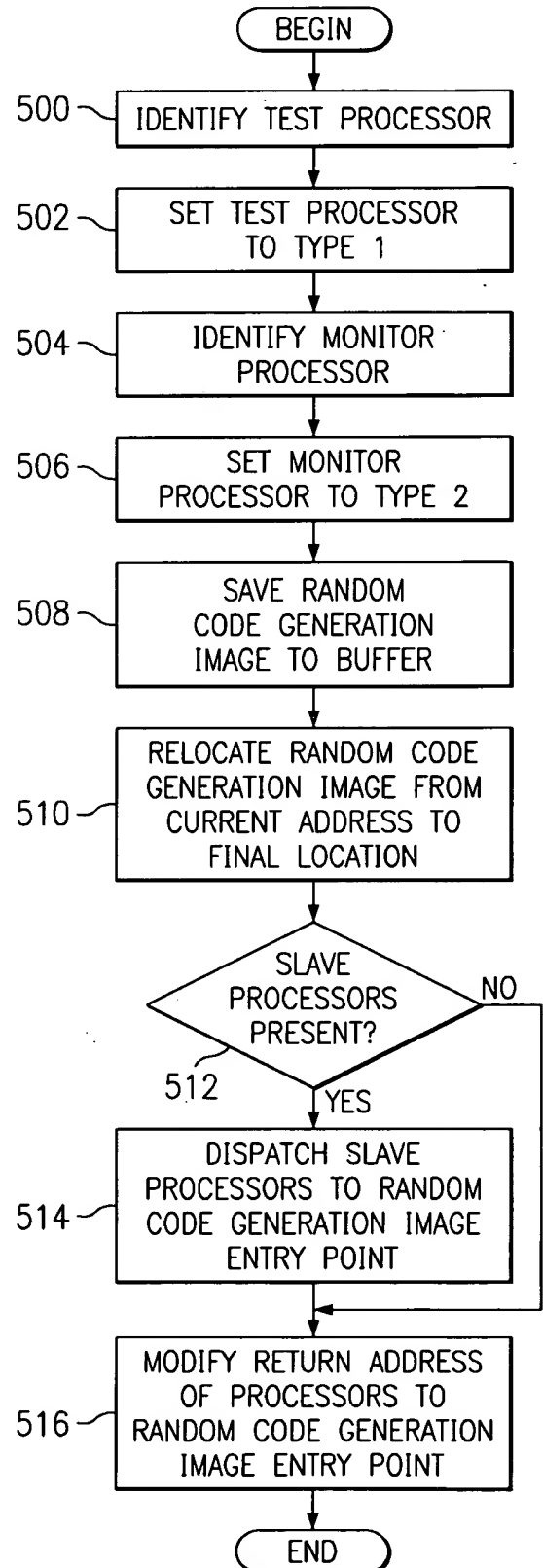


FIG. 6

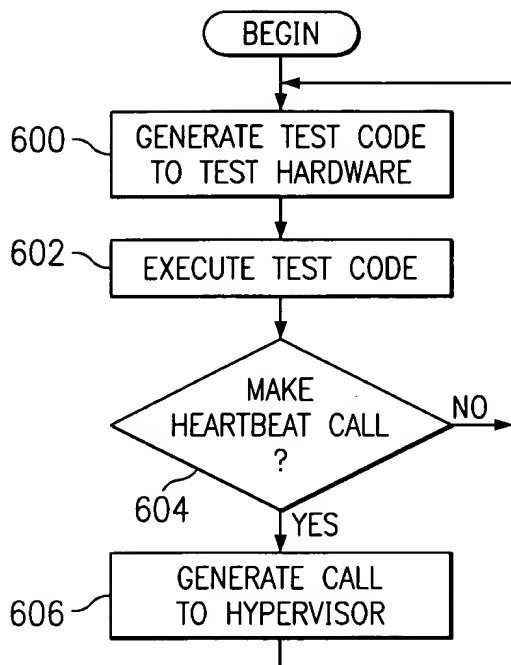


FIG. 8

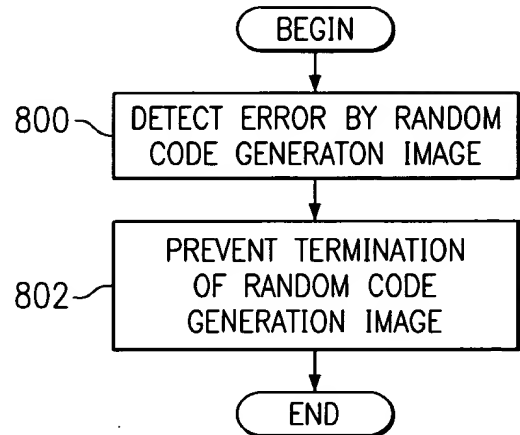
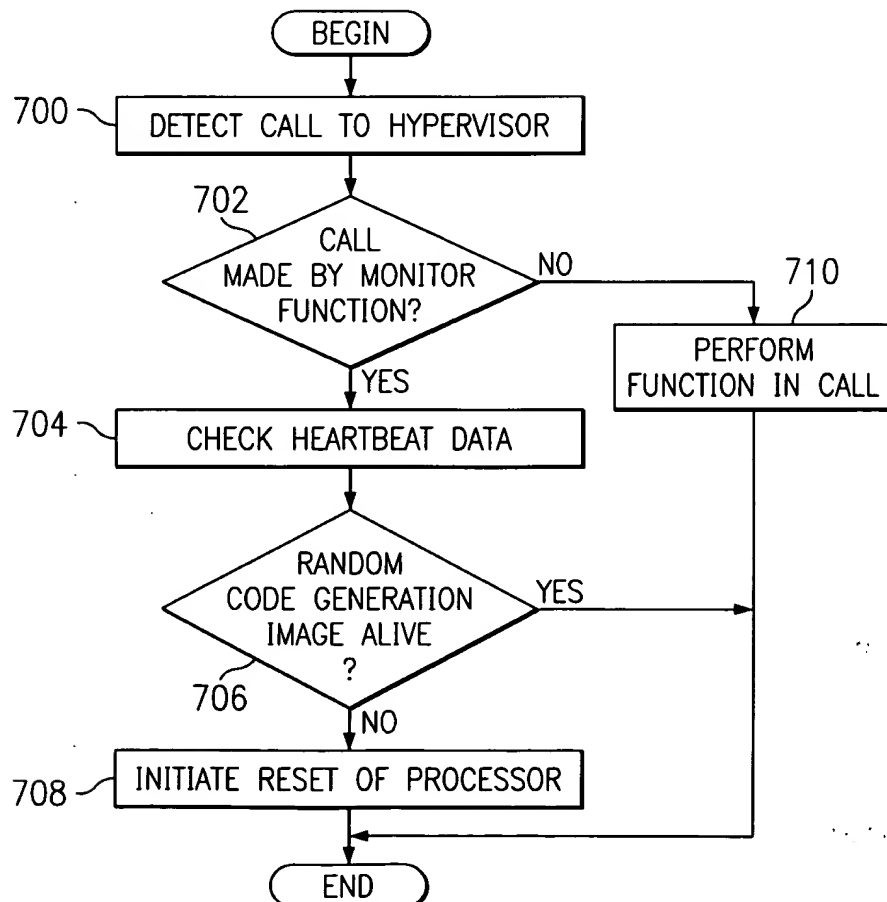


FIG. 7



```

struct          902
{
    unsigned long heartbeat = 0;      /* increment one each time by the LTK processor making the 904
    "heartbeat" h-call */
    unsigned long last_heartbeat = -1; /* capture the heartbeat into this field by the MONITOR processor */
    unsigned long timestamp = -1;     /* time stamp when the capture takes place if there is still
    heartbeat ticking */
} heartbeat[MAX_PROCS]; 906

```

FIG. 9

```

h_heartbeat()
{
    my_proc_id = get_proc_id();
    my_part_id = get_part_id();
    if ( my_part_id == GLOBAL_FW ) return; /* can't call from global FW */
    if ( proc_type[ my_proc_id ] != LTK_TYPE ) return; /* processor is not running LTK */
    else
    {
        extern unsigned long refresh_frequency;
        extern PART_INFO part_info[];
        unsigned long newtime;

        ++heartbeat[ my_proc_id ].heartbeat;
        simple_lock( &part_info[ my_part_id ].refresh_lock ); /* serialize access to
        last_refresh_timestamp */
        newtime = get_current_time_ticks();
        if ( (newtime - part_info[ my_part_id ].last_refresh_timestamp) > refresh_frequency )
        {
            /* time is up for a image refresh */
            restore_LTK_img_from_saved_copy();
            part_info[ my_part_id ].last_refresh_timestamp = get_current_time_ticks();
        }
        simple_unlock( &part_info[ my_part_id ].refresh_lock );
    }
}

```

FIG. 10

```

monitor_code()
{
    my_proc_id = get_proc_id();
    my_sister_id = get_sister_id( my_proc_id );
    if ( proc_type[ my_proc_id ] == MONITOR_TYPE )
    {
        if ( heartbeat[ my_sister_id ].last_heartbeat != heartbeat[ my_sister_id ].heartbeat )
        {
            /* there is heartbeat ticking, capture the new heartbeat and record new timestamp */
            heartbeat[ my_sister_id ].last_heartbeat = heartbeat[ my_sister_id ].heartbeat;
            heartbeat[ my_sister_id ].timestamp = get_current_time_ticks();
        }
        else
        {
            extern unsigned long heartbeat_frequency;
            unsigned long newtime;

            /* if there is no heartbeat which exceeds the required heartbeat observed interval */
            /* initiate a reset to the LTK processor monitored by this processor */
            newtime = get_current_time_ticks();
            if ( (newtime - heartbeat[ my_sister_id ].timestamp) > heartbeat_frequency )
                reset_the_sister_LTK_processor();
        }
    }
}

```

FIG. 11

FIG. 12

monitor\_glue\_code: # at a fixed location in hypervisor memory  
 bir # Simply return  
 b monitor\_code # Call the actual monitor code

1200